S&H Form: PTO/SB/05 (8/97)

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	1046.1210/JDH
First Named Inventor or Application Identifier:	Yoshinobu NAKAMURA
Express Mail Label No.	

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: **Assistant Commissioner for Patents**
Box Patent Application
Washington, DC 20231

1. [X] Fee Transmittal Form
2. [X] Specification, Claims & Abstract ..... [Total Pages: 16]
3. [X] Drawing(s) *(35 USC 113)* ..... [Total Sheets: 5]
4. [X] Oath or Declaration ..... [Total Pages: 3]
   a. [X] Newly executed (original or copy)
   b. [ ] Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional with Box 17 completed)*
      i. [ ] DELETION OF INVENTOR(S)
      Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
5. [ ] Incorporation by Reference (usable if Box 4b is checked)
   The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.
6. [ ] Microfiche Computer Program *(Appendix)*
7. [ ] Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
   a. [ ] Computer Readable Copy
   b. [ ] Paper Copy (identical to computer copy)
   c. [ ] Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. [X] Assignment Papers (cover sheet & document(s))
9. [ ] 37 CFR 3.73(b) Statement *(when there is an assignee)* [ ] Power of Attorney
10. [ ] English Translation Document *(if applicable)*
11. [X] Information Disclosure Statement (IDS)/PTO-1449[X] Copies of IDS Citations
12. [ ] Preliminary Amendment
13. [X] Return Receipt Postcard (MPEP 503) *(Should be specifically itemized)*
14. [ ] Small Entity Statement(s) [ ] Statement filed in prior application, status still proper and desired.
15. [X] Certified Copy of Priority Document(s) *(if foreign priority is claimed)*
16. [ ] Other:

17. If a CONTINUING APPLICATION, *check appropriate box and supply the requisite information:*

[ ] Continuation [ ] Divisional [ ] Continuation-in-part (CIP) of prior application No: ___/_____

18. CORRESPONDENCE ADDRESS

STAAS & HALSEY, LLP
Attn: James D. Halsey, Jr.
700 Eleventh Street, N.W., Suite 500
Washington, DC 20001

Telephone: (202) 434-1500
Facsimile: (202) 434-1501

LABEL ADDRESS TRANSLATING DEVICE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a label address translating device for obtaining an effective address (an absolute address, i.e., an intra-memory actual address from a label in a program described in a predetermined language.

2. Related Background Art

In a program described in a computer language such as a C language, a BASIC language, a LIPS language and an APL language, a label address translation for obtaining an effective address from a label is executed with reference to a table showing a relationship between the label and the effective address.

For example, Japanese Patent Application Laid-Open Publication No.1-258029 discloses a data processor including a table showing a correspondence between a symbol string (labels) indicating respective identification names with respect to data areas and programs, and storage addresses in a storage device. This data processor obtains the storage address of the program or the data area from the correspondence table, and thus executes the program.

Further, in a program described in an interpreter language of being translated and interpreted per text, a label table showing a relationship between the label and the address is referred to each time the program is executed, and there are obtained effective addresses of a function and a variable which correspond to the label.

Moreover, in a program described in a compiler language in which all the texts are compiled and execution is thereafter done, the effective address is obtained from the label table each time the program is compiled.

The program described in the interpreter language, however, has such a problem that the label table is referred to with an interruption of the fundamental process during the execution of the program, and therefore a speed of executing the program might be delayed.

A problem inherent in the program described in the compiler language is that the compilation of the program takes much time.

A problem of a program described in a language containing the interpreter and compiler languages is that there might be a scatter in the execution speed.

SUMMARY OF THE INVENTION

It is a primary object of the present invention, which was devised to obviate the problems described above, to provide a label address translating device capable of increasing an executing speed with respect to a program described in an interpreter language and a compiling speed with respect to a program described in a compiler language, and preventing a scatter in the executing speed with respect to a program described in a language (e.g., a Java language) containing both of the interpreter language and the compiler language.

To accomplish the above object, according to one aspect

of the present invention, a label address translating device for obtaining an address from a label described in a program, comprises a program processing unit judging whether or not the address is obtained from the label when in the processing of the program, and a label address translating unit obtaining, when the program processing unit judges that the address is not obtained from the label, the address from the label in an exception handling, and writing the address to the program.

According to the present invention, when the program processing unit judges that the address is not obtained from the label during the execution of the program, the label address translating unit obtains the address from the label and write the obtained address to the program. Thus, the address obtained from the label is written to the program, and thereafter the program to which the address has been written is processed (compiled). Hence, there is no necessity for acquiring the address from the label each time the program is processed (compiled).

Accordingly, if translating once the program described in the interpreter language, there is no necessity for obtaining the address from the label with an interruption of the fundamental process during the execution of the program. It is therefore feasible to increase the speed of executing the program. Further, as for the program described in the compiler language, the program compiling speed can be raised. Moreover, the scatter in the executing speed does not occur in the program described in the language containing both of the interpreter

language and the compiler language.

Herein, the program processing unit and the label address translating unit may be constructed as functions actualized by, for example, a CPU executing the program for obtaining the address from the label. For instance, a main memory of a computer is stored with an exception handler, and this exception handler is executed by the CPU, thereby actualizing the label address translating unit.

The label address translating device according to the present invention may be constructed so that the label address translating unit may use a table showing a relationship between the label and the address.

With this contrivance, the address relative to the label can be obtained by retrieving the table with a label name and a label ID serving as a key.

The table is stored in, e.g., the main memory of the computer and accessed by the CPU.

The label address translating device according to the present invention may be constructed so that the program processing unit, when the address indicated by the label is not a real address, judges that the address is not obtained from the label.

Herein, the effective address may include an absolute address, an actual address in the memory, etc. but excludes an address inaccessible directly by the CPU.

According to another aspect of the present invention, a label address translating method of obtaining an address from

a label described in a program, comprises a first step of processing the program, a second step of judging whether or not the address is obtained from the label when the processing the program in the first step, a third step of obtaining, when judging in the second step that the address is not obtained from the label, the address from the label in an exception handling, and a fourth step of writing the address obtained in the third step to the program.

The label address translating method according to the present invention may be structured to that the third step involves reading the address to be obtained from a table showing a relationship between the label and the address.

The label address translating method according to the present invention may be structured so that the second step involves judging, when the address indicated by the label is not a real address, that the address is not obtained from the label.

According to a further aspect of the present invention, there is provided a readable-by-computer medium stored with a second program for making a computer function as a label address translating device for obtaining an address from a label described in a first program. The second program comprises a judging step of judging whether or not the address is obtained from the label when processing the first program, an obtaining step of obtaining, when judging in the judging step that the address is not obtained from the label, the address from the label in an exception handling, and a writing step of writing

the address obtained in the obtaining step to the program.

The readable-by-computer medium may embrace, e.g., a CD-ROM, a magnetic disk such as a floppy disk, a magneto-optic disk such as an MO, and an optical disk such as a PD.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagram showing an architecture of a label address translating device in an embodiment of the present invention;

FIG. 2 is a diagram showing one example of a source code of a program shown in FIG. 1;

FIG. 3 is a diagram showing one example of a code obtained by a compilation of the program;

FIG. 4 is a diagram showing one example of a label table shown in FIG. 1;

FIG. 5 is a diagram showing one example of a code in which the address is rewritten by a label address translating unit illustrated in FIG. 1;

FIG. 6 is a flowchart showing a process by the label address converting device illustrated in FIG. 1; and

FIG. 7 is a flowchart showing an exception handling shown in FIG. 6.

DESCRIPTION OF THE PREFERRED EMBODIMENT

An embodiment of the present invention will hereinafter be described with reference to the accompanying drawings.

<Architecture of Label Address Translating Device>

To start with, an architecture of a label address translating device in the embodiment of the present invention will be explained referring to FIGS. 1 through 5. FIG. 1 is a diagram showing a construction of a label address translating device 10. FIG. 2 is a diagram showing one example of a source code of a program 20 shown in FIG. 1. FIG. 3 is a diagram showing one example of a code obtained by compilation of the program. FIG. 4 is a diagram showing one example of a label table 24 illustrated in FIG. 1. FIG. 5 is a diagram showing one example of a code in which an address is rewritten by a label address translating unit 18 shown in FIG. 1.

As illustrated in FIG. 1, the label address translating device 10 is actualized by, e.g., a personal computer executing a label address translation program. The label address translating device 10 includes a CPU 12 and a main memory 14 accessed directly by the CPU 12.

The CPU 12 executes the label address translation program stored in a storage unit such as a ROM a hard disk and a magnetic disk (none of those storage devices is shown herein, and the storage device corresponds to a readable-by-computer medium according to the present invention), thereby functioning as a label address translating unit 18 (corresponding to a label address translating module) as well as a program executing unit 16 (corresponding to a program processing module).

Further, with the execution of the label address translation program by the CPU 12, an application program 20 as a processing target, an exception handler 22, a label table

32 (corresponding to a table) showing a relationship between the label and the address and a compiler 26, are developed on the main memory 14.

A source code 28 of the program 20 is described by use of a label name (LABEL_B) in a predetermined language, e.g., an interpreter language (see FIG. 2). Further, the source code 28 is encoded by an interpreter contained in the source code 28 (see FIG. 3). The encoded program 20 is developed on the main memory 14 and executed by the CPU 12.

In a code 30 shown in FIG. 3, [FLD 10] and [FLD 20] indicate loading of "10" and "20", [FADD] indicates adding loaded "10" and "20", and [FSTP DWORD PTR [0xFFFFFFFE]] indicates storing an added result in an address "0xFFFFFFFE". Moreover, the invalid data "0xFFFFFFFE" as the address in which the added result should be stored, i.e., the data ruled out of an accessible memory range when executing the program 20, is defined in the code 30. The CPU 12 is, when detecting the invalid data, set to execute an exception handling. The exception handling has a higher execution speed than a normal program execution process, and is executed in advance of the fundamental execution process.

The label table 32 is, as shown in FIG. 1 or 4, stored with label Ids, effective addresses that should be referred to by a command, and label names described in the source code 28. The label table 32 is created during a compilation of the program.

Note that what is typical as a program edit tool of the

interpreter language at the present is that an intermediate code is compiled for the duration of a compilation of a source program by the user in order to increase an processing speed in a posterior execution process. The compile process is carried out during the editing of the source program also in the edit tool of the interpreter language in this embodiment.

The label table 32 shows a corresponding relationship between the label ID "0xFFFFFFFE" defined as the address in the code shown in FIG. 3, the effective address "0x00050006" and the label name "LABEL B". That is, the (effective) address can be obtained by referring to the label table 32, wherein the label name and the label ID described in the program 20 serve as a key. A method of an address modification of the effective address may be based on either absolute addressing or relative addressing.

Referring back to FIG. 1, the program execution unit 16 executes the program 20 developed on the main memory 14 in accordance with an execution command inputted by an operator.

Further, the program execution unit 16 executes a process of referring to the address during the execution of the program 20. For instance, the program execution unit 16 performs a certain arithmetic operation and stores an arithmetic result thereof in a predetermined address of the memory. At this time, the program execution unit 16, when recognizing that an invalid piece of data (invalid data) is described in the program 20, the exception handling (i.e., the label address translation process) other than the normal processes of the program 20.

The label address translating unit 18, when executing the exception handling, starts up an exception handling handler 22. The exception handler 22 retrieves the label table 32, in which an address of a command (e.g., [FSTP]) becoming a target for the exception handling and the label described in the source code 28 of the program, serve as a key. The exception handler 22 thus reads the relevant effective address from the label table 32.

Thereafter, the label address translating unit 18 sets the obtained effective address in the program 20. For example, [FSTP DWORD PTR[0xFFFFFFFE]] shown in FIG. 3 is, as shown in FIG. 5, rewritten into [FSTP DWORD PRT[0x00050006]]. This piece of data "0x00050006" indicates an effective address of the memory.

According to the label address translating device 10, when recognizing the invalid data during the execution of the program 20 by the program execution unit 16, the label address translating unit 18 starts up the exception handler 22. Subsequently, the exception handler 22 obtains the effective address from the label defined in the program 20 by referring to the label tables 24, 32, and sets the thus obtained effective address in the program 20. Namely, the CPU 12, upon recognizing the invalid data, executes the exception handling and the normal processes thereafter.

When the effective address is thereby set in a command (such as storing the arithmetic result in the predetermined address on the memory) of referring to the address, thereafter,

the exception handling for translating the label into the address is not required to be done with respect to that command. Accordingly, it is feasible to speed up the execution of the program 20 and the translation as well.

Further, the process of obtaining the effective address from the label ID is carried out as an exception handling by the CPU 12 (particularly by the program executing unit 16). The exception handling is executed by a method different from the normal process at a higher execution speed than the normal process. Hence, a translation speed can be made higher than in the case of implementing the label address translation process in the same way as the normal process.

<Operational Example of Label Address Translating Device>

Next, an operational example of the label address translating device 10 described above will be explained. FIG. 6 is a flowchart showing the processes by the label address translating device 10 illustrated in FIG. 1. FIG. 7 is a flowchart of the exception handling shown in FIG. 6.

As shown in FIG. 6, upon a start of executing the program 20 (S01), the CPU 12 implements the command described in the program 20 (S02). At that time, the CPU 12, when recognizing the invalid data, executes the label address translating process as the exception handling (S03).

As shown in FIG. 7, the CPU 12, upon recognizing the invalid data during the execution of the program 20, starts up the exception handler 22 (S11). The exception handler 22 refers

to the label tables 24, 32, thereby obtaining the effective address, wherein the address of the command in which the invalid data is described (or the label ID of this command) serves as a key (S12). Subsequently, the CPU 12 rewrites the address described in that command (which is, e.g., an address for storing the arithmetic result) into the effective address from the invalid data (S13).

Thereafter, the CPU 12 sets an execution pointer (viz., the process) back to the command by which the exception handling is recognized (S14). The address described in that command has been rewritten into the effective address from the invalid data, and hence the CPU 12 executes the command based on the rewritten address (for instance, stores the arithmetic result in a predetermined address), and further executes commands thereafter (S15).

According to the label address translating device 10 in the embodiment discussed above, the label tables 24, 32 showing the relationships between the labels and the effective addresses are referred to, whereby each of the labels described in the program 20 is rewritten into the effective address. The program 20 as a target for processing eventually falls into a state where the effective address is rewritten therein instead of the label. Hence, in the execution (compilation) of the program 20 after being rewritten, since there is no necessity for translating the label into the effective address, it is possible to omit the process of translating the label into the effective address.

Accordingly, if the program 20 is described in the interpreter language, the speed of executing the program 20 can be increased. If the program 20 is described in the compiler language, a translation speed of the program 20 can be increased. If the program 20 is described in a language (e.g., Java language) containing both of the interpreter language and the compiler language, it is feasible to restrain a scatter in the execution speed between the commands contained in the program 20.

WHAT IS CLAIMED IS:

1. A label address translating device for obtaining an address from a label described in a program, comprising:

a program processing unit judging whether or not the address is obtained from the label when in the processing of the program; and

a label address translating unit obtaining, when said program processing unit judges that the address is not obtained from the label, the address from the label in an exception handling, and writing the address to the program.

2. A label address translating device according to claim 1, wherein said label address translating unit uses a table showing a relationship between the label and the address.

3. A label address translating device according to claim 1, wherein said program processing unit, when the address indicated by the label is not a real address, judges that the address is not obtained from the label.

4. A label address translating method of obtaining an address from a label described in a program, comprising:

a first step of processing the program;

a second step of judging whether or not the address is obtained from the label when the processing the program in said first step;

a third step of obtaining, when judging in said second

step that the address is not obtained from the label, the address from the label in an exception handling; and

a fourth step of writing the address obtained in said third step to the program.

5. A label address translating method according to claim 4, wherein said third step involves reading the address to be obtained from a table showing a relationship between the label and the address.

6. A label address translating method according to claim 4, wherein said second step involves judging, when the address indicated by the label is not a real address, that the address is not obtained from the label.

7. A readable-by-computer medium stored with a second program for making a computer function as a label address translating device for obtaining an address from a label described in a first program,

said second program comprising:

a judging step of judging whether or not the address is obtained from the label when processing said first program;

an obtaining step of obtaining, when judging in said judging step that the address is not obtained from the label, the address from the label in an exception handling; and

a writing step of writing the address obtained in said obtaining step to said program.

ABSTRACT OF THE DISCLOSURE

A label address translating device includes a program processing unit for processing a program and judging, when processing the program, whether or not an address is obtained from a label, and a label address translating unit for reading the address corresponding to the label from a label table in an exception handling, and writing the address to the program. the program after being processed by the label address translating device comes into a state where the label is rewritten into the address. The rewriting into the address is performed in the exception handling, and a time needed for processing (compiling) the program can therefore be reduced.

FIG. 1

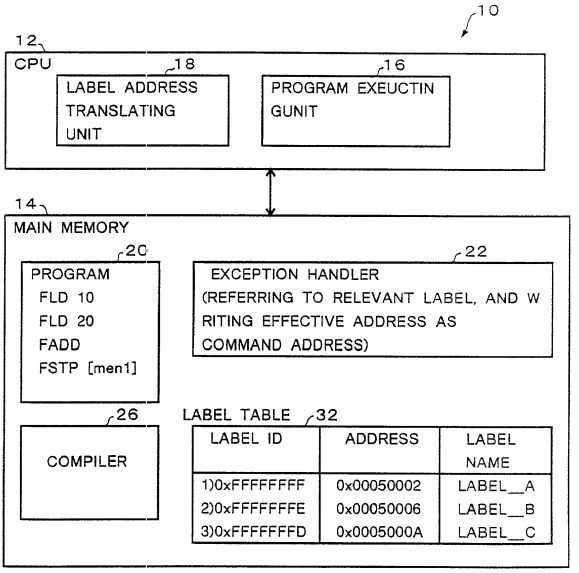

10
12
CPU
18
LABEL ADDRESS TRANSLATING UNIT
16
PROGRAM EXEUCTIN GUNIT
14
MAIN MEMORY
20
PROGRAM
FLD 10
FLD 20
FADD
FSTP [men1]
22
EXCEPTION HANDLER
(REFERRING TO RELEVANT LABEL, AND W
RITING EFFECTIVE ADDRESS AS
COMMAND ADDRESS)
26
COMPILER
LABEL TABLE
32
LABEL ID
ADDRESS
LABEL NAME
1)0xFFFFFFFF
0x00050002
LABEL__A
2)0xFFFFFFFE
0x00050006
LABEL__B
3)0xFFFFFFFD
0x0005000A
LABEL__C

FIG. 2

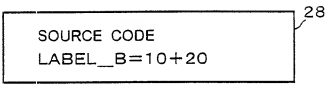

28
SOURCE CODE
LABEL_B=10+20

FIG. 3

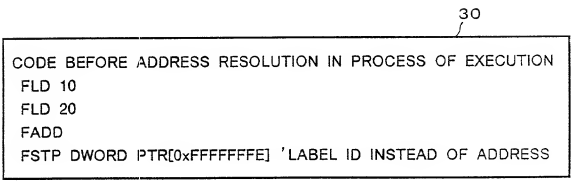

30
CODE BEFORE ADDRESS RESOLUTION IN PROCESS OF EXECUTION
FLD 10
FLD 20
FADD
FSTP DWORD PTR[0xFFFFFFFE] 'LABEL ID INSTEAD OF ADDRESS

FIG. 4

32

LABEL ID	ADDRESS	LABEL NAME
1)0xFFFFFFFF	0x00050002	'LABEL__A
2)0xFFFFFFFE	0x00050006	'LABEL__B
3)0xFFFFFFFD	0x0005000A	'LABEL__C

FIG. 5

```
CODE BEFORE ADDRESS RESOLUTION IN PROCESS OF EXECUTION
  FLD 10
  FLD 20
  FADD
  FSTP DWORD PTR[0x00050006]
```

FIG. 6

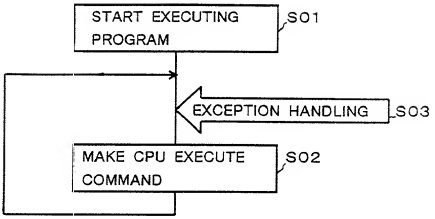
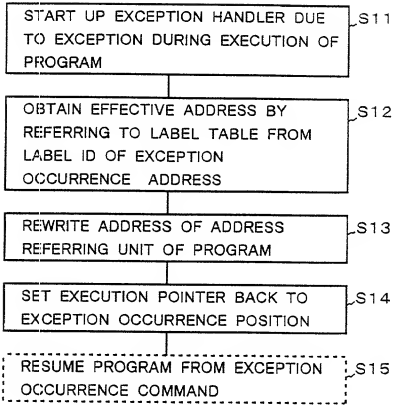

START EXECUTING PROGRAM
S01
EXCEPTION HANDLING
S03
MAKE CPU EXECUTE COMMAND
S02

FIG. 7


START UP EXCEPTION HANDLER DUE TO EXCEPTION DURING EXECUTION OF PROGRAM
S11
OBTAIN EFFECTIVE ADDRESS BY REFERRING TO LABEL TABLE FROM LABEL ID OF EXCEPTION OCCURRENCE ADDRESS
S12
REWRITE ADDRESS OF ADDRESS REFERRING UNIT OF PROGRAM
S13
SET EXECUTION POINTER BACK TO EXCEPTION OCCURRENCE POSITION
S14
RESUME PROGRAM FROM EXCEPTION OCCURRENCE COMMAND
S15

PTO/SB/105 (8-96)
Approved for use through 9/30/98. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Declaration and Power of Attorney For Patent Application

特許出願宣言書及び委任状

Japanese Language Declaration

日本語宣言書

下記の氏名の発明者として、私は以下の通り宣言します。	As a below named inventor, I hereby declare that:
私の住所、私書箱、国籍は下記の私の氏名の後に記載された通りです。	My residence, post office address and citizenship are as stated next to my name.
下記の名称の発明に関して請求範囲に記載され、特許出願している発明内容について、私が最初かつ唯一の発明者（下記の氏名が一つの場合）もしくは最初かつ共同発明者であると（下記の名称が複数の場合）信じています。	I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled
_____	LABEL ADDRESS TRANSLATING DEVICE
_____	_____
上記発明の明細書（下記の欄でx印がついていない場合は、本書に添付）は、	the specification of which is attached hereto unless the following box is checked:
☐ __月__日に提出され、米国出願番号または特許協定条約国際出願番号を_____とし、（該当する場合）_____に訂正されました。	☐ was filed on _____ as United States Application Number or PCT International Application Number _____ and was amended on _____ (if applicable).
私は、特許請求範囲を含む上記訂正後の明細書を検討し、内容を理解していることをここに表明します。	I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.
私は、連邦規則法典第３７編第１条５６項に定義されるとおり、特許資格の有無について重要な情報を開示する義務があることを認めます。	I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

Japanese Language Declaration
(日本語宣言書)

私は、米国法典第３５編１１９条(a)-(d)項又は３６５条(b)項に基と下記の、米国以外の国の少なくとも一ヵ国を指定している特許協力条約３６５(a)項に基づく国際出願、又は外国での特許出願もしくは発明者証の出願についての外国優先権をここに主張するとともに、優先権を主張している、本出願の前に出願された特許または発明者証の外国出願を以下に、枠内をマークすることで、示しています。

I hereby claim foreign priority under Title 35, United States Code, Section 119 (a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)
外国での先行出願

(Number) (番号)	(Country) (国名)	(Day/Month/Year Filed) (出願年月日)	Priority Not Claimed 優先権主張なし
11-041430	Japan	19/02/1999	☐
12-9955	Japan	13/01/2000	☐

私は、第３５編米国法典１１９条(e)項に基いて下記の米国特許出願規定に記載された権利をここに主張いたします。

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below.

(Application No.) (出願番号)	(Filing Date) (出願日)

(Application No.) (出願番号)	(Filing Date) (出願日)

私は、下記の米国法典第３５編１２０条に基いて下記の米国特許出願に記載された権利、又は米国を指定している特許協力条約３６５条(c)に基づく権利をここに主張します。また、本出願の各請求範囲の内容が米国法典第３５編１１２条第１項又は特許協力条約で規定された方法で先行する米国特許出願に開示されていない限り、その先行米国出願書提出日以降で本出願書の日本国内または特許協力条約国際提出日までの期間中に入手された、連邦規則法典第３７編１条５６項で定義された特許資格の有無に関する重要な情報について開示義務があることを認識しています。

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s), or 365(c) of any PCT international application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of application.

(Application No.) (出願番号)	(Filing Date) (出願日)	(Status: Patented, Pending, Abandoned) (現況：特許許可済、係属中、放棄済)
(Application No.) (出願番号)	(Filing Date) (出願日)	(Status: Patented, Pending, Abandoned) (現況：特許許可済、係属中、放棄済)

私は、私自身の知識に基づいて本宣言書中で私が行なう表明が真実であり、かつ私の入手した情報と私の信じるところに基づく表明が全て真実であると信じていること、さらに故意になされた虚偽の表明及びそれと同等の行為は米国法典第１８編第１００１条に基づき、罰金または拘禁、もしくはその両方により処罰されること、そしてそのような故意による虚偽の声明を行なえば、出願した、又は既に許可された特許の有効性が失われることを認識し、よってここに下記のごとく宣誓を致します。

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

PTO/SB/106 (6-95)
Approved for use through 6/30/96. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Japanese Language Declaration
(日本語宣言書)

委任状： 私は下記の発明者として、本出願に関する一切の手続きを米特許商標局に対して遂行する弁理士または代理人として、下記の者を指名いたします。（弁護士、または代理人の氏名及び登録番号を明記のこと）

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith *(list name and registration number)*

James D. Halsey, Jr., 22,729; Harry John Staas, 22,010; David M. Pitcher, 25,908; John C. Garvey, 28,607; J. Randall Beckers, 30,358; William F. Herbert, 31,024; Richard A. Gollhofer, 31,106; Mark J. Henry, 36,162; Gene M. Garner II, 34,172; Michael D. Stein, 37,240; Paul I. Kravetz, 35,230; Gerald P. Joyce, III, 37,648; Todd E. Marlette, 35,269; Harlan B. Williams, Jr., 34,756; George N. Stevens, 36,938; Michael C. Soldner, 41,455; Norman L. Ourada, 41,235; Kevin R. Spivak, P-43,148; and William M. Schertler, 35,348 (agent)

書類送付先

Send Correspondence to:

STAAS & HALSEY
700 Eleventh Street, N.W.
Suite 500
Washington, D.C. 20001

直接電話連絡先： (名前及び電話番号)

Direct Telephone Calls to: *(name and telephone number)*

STAAS & HALSEY
(202) 434-1500

単一または第一発明者名		Full name of sole or first inventor: Yoshinobu NAKAMURA
発明者の署名	日付	Inventor's signature: Yoshinobu Nakamura Date: February 2, 2000
住所		Residence: Inagi-shi, Tokyo, Japan
国籍		Citizenship: Japanese
私書箱		Post Office Address: c/o FUJITSU PERSONAL COMPUTER SYSTEMS LIMITED, 1405 Ooaza Oomaru, Inagi-shi, Tokyo, Japan
第二共同発明者		Full name of second joint inventor, if any
第二共同発明者	日付	Second inventor's signature / Date
住所		Residence
国籍		Citizenship
私書箱		Post Office Address

（第三以降の共同発明者についても同様に記載し、署名をすること）

(Supply similar information and signature for third and subsequent joint inventors.)